Manual $\LaTeX$

Carlos Roberto Córdoba
UFMS

Doglas Wendll Sorgatto
UFMS

2 de Maio de 2023

Conteúdo

1 Introdução

Atualmente é preciso fazer uso de um computador em todas as áreas que imaginarmos, independente se é usado de forma direta, ou indireta, ele está lá. Uma das funções que é primordial para qualquer pessoa, é a edição de textos, seja para confeccionar um trabalho, um relatório, ou um simples lembrete um pouco mais exigente. O usuário precisa escolher um tamanho de fonte adequado, o estilo, a cor, etc. A edição não para por aí, o que as pessoas precisam editar não é somente como a fonte está sendo apresentada, mas também, como ela é apresentada no meio em que se encontra, neste caso, a representação da página, incluindo configurações tais como o tamanho da folha e as medidas das margens.

Para tal tarefa, geralmente, os usuários usam soluções de software um pouco mais sofisticadas em relação àquelas que estão presentes nativamente em seu sistema operacional. Exemplos mais conhecidos dessas soluções são o `Microsoft Office` (representando uma suíte de aplicativos de escritório paga), e o `LibreOffice` (representando uma solução livre, e consequentemente gratuita). Nesses aplicativos, os usuários veem exatamente o que será impresso na mesma hora em que mudam alguma configuração de edição, classificando esses softwares como **Editores Visuais**.

O objetivo deste texto é apresentar a **Edição Lógica**, ou seja, uma forma de edição de texto que garante qualidade, facilidade em trabalhar com fórmulas matemáticas, bibliografias, citações, índices e a consistência de sumários, listas de tabelas e de figuras, além de permitir gerenciamento inteligente de notas de rodapé. A apresentação do texto possui uma qualidade surpreendente, tanto na tela, quanto no momento da impressão. Apesar de todas essas qualidades e de muitas outras, promover a utilização da Edição Lógica pode não ser tão simples e, em muitos casos, não representa nenhum ganho para o usuário que trabalha com uma edição mais básica, em que as ferramentas visuais são muito mais intuitivas para ele.

Na Edição Lógica, o texto é customizado por meio de comandos embutidos no que denominamos de arquivo fonte. Esse arquivo fonte não mostra visualmente a maneira como o texto está editado, para visualizar o resultado, é preciso submetê-lo a um processo chamado compilação, gerando um arquivo de saída que de fato apresenta o texto editado (geralmente em pdf).

Abordaremos neste minicurso o processo de edição de textos utilizando o Editor Lógico `Texmaker` (e, consequentemente, um conjunto de comandos LaTeX), pelo fato de ter uma boa interface e ser multiplataforma. Além do editor, é preciso ter instalado um compilador LaTeX, detalhes de como preparar o ambiente para compilação de LaTeX, serão detalhados no decorrer deste manual.

2 Instalando LATEX

O primeiro passo para começar a utilizar o LATEXé instalar o conjunto de pacotes que permitem a compilação do arquivo fonte. Em distribuições `Linux` baseadas no Debian, é possível obter esses pacotes por meio do `apt-get`. Para essa tarefa, abra um terminal, e execute esse comando (copie a linha, cole no interior do terminal, e pressione a tecla enter, geralmente será solicitada a senha de root somente uma vez, caso tenha dúvidas a respeito dessa senha, contate o administrador do sistema):

- `sudo apt-get install texlive texlive-latex-extra texlive-lang-portuguese`

Feito isso, já é possível compilar LATEX, mas neste manual não abordaremos a compilação do arquivo fonte por meio de um terminal, o próprio editor que escolhemos fará essa tarefa. Esse é próximo passo, devemos instalar o editor `Texmaker`, que pode ser instalado da mesma forma que os pacotes anteriores:

- `sudo apt-get install texmaker`.

Vale lembrar que caso esteja utilizando o Debian, e não suas derivações, provavelmente não será possível usar o comando "sudo" (que tem a função de conceder privilégios de superusuário, necessário para instalação de aplicativos em ambiente `Unix`). Dada essa restrição, no Debian devemos utilizar o comando "su", que possui a mesma função que o sudo, mas não expira os privilégios, sendo concedidos por tempo indeterminado, enquanto o usuário não encerra o terminal. Assim, devemos rodar o comando "su" em um terminal, digitar a senha de root, e executar os mesmos comandos anteriores, sem o comando "sudo", dessa forma, rodaremos os comandos:

- `apt-get install texlive texlive-latex-extra texlive-lang-portuguese`
- `apt-get install texmaker` (para instalar o editor).

Para usuários `Windows`, a tarefa é muito mais simples, sendo necessários somente baixar os instaladores e executá-los. Versões posteriores ao Windows XP será solicitada uma confirmação para executar uma ação privilegiada. Caso o usuário seja protegido por senha, essa senha será solicitada. Outro fator que facilita o uso em ambiente `Windows`, é o tamanho do arquivo de instalação e conforme há a necessidade de usar um pacote específico, este é baixado automaticamente.

Assim como na preparação para o ambiente `Linux`, a primeira parte da instalação é referente aos pacotes necessários para compilar o arquivo fonte. Como descrito anteriormente, aqui não usaremos o terminal (geralmente denominado `prompt` de comando no `Windows`) para compilação, esse processo é automatizado pelo editor. Basta acessar o endereço: http://miktex.org/, baixar o executável adequado. O instalador é muito simples, não necessitando de maiores detalhes. Instalado esse pacote, iremos instalar o `Texmaker`, seguindo o mesmo processo, baixando o instalador através deste link: http://www.xm1math.net/texmaker/

3 Criando documentos - Documentclass

O primeiro passo na hora de criar um novo documento em LaTeX, é definir a que "classe" esse documento pertence, faremos isso através do comando `\documentclass`. Entre esse comando e o `\begin{document}`, (usado para determinar onde o nosso documento tem início), temos o que chamamos de "preâmbulo", área na qual todos os comandos são aplicados ao documento inteiro. A Tabela 1 apresenta alguns exemplos de "classes" disponíveis.

Tabela 1: Exemplos de classes.

Classe	**Aplicação**
article	Para artigos em revistas científicas, apresentações, relatórios curtos, documentação de programas, convites, etc.
IEEEtran	Para artigos com o formato IEEE Transactions.
proc	A classe de processo com base na classe artigo.
minimal	Define um tamanho de página e uma fonte de base. É usado principalmente para fins de depuração.
report	Para relatórios mais longos contendo vários capítulos, pequenos livros, teses, etc.
book	Para livros.
slides	Para slides. A classe usa grandes letras sem serifa.
letter	Para escrever cartas.
beamer	Para escrever apresentações.

Exemplo de uso: `\documentclass{x}`, onde "x" é o nome da classe.

Além de passar como parâmetro no comando `\documentclass{}` a classe escolhida, podemos definir algumas opções adicionais para personalizar o documento. Para isso, devemos usar colchetes antes das chaves. A Tabela 2 apresenta algumas dessas opções.

Tabela 2: Exemplos de opções do comando `\documentclass{x}`

Opção	**Aplicação**
10pt, 11pt, 12pt	Define o tamanho da fonte principal no documento. Se nenhuma opção for especificada, é assumido 10pt.
a4paper, letterpaper	Define o tamanho do papel. O tamanho padrão é **letterpaper**; No entanto, muitas distribuições européias de Tex já vêm pré-definido para A4, não carta, e isso também é verdade para todas as distribuições do pdflatex. Além disso, **a5paper**, **b5paper**, **executivepaper**, e **legalpaper** podem ser especificados.
fleqn	Edita fórmulas apresentadas alinhando à esquerda em vez de centralizado.
leqno	Coloca a numeração das fórmulas do lado esquerdo, em vez de à direita.
titlepage, notitlepage	Especifica se uma nova página deve ser iniciada após o título do documento ou não. A classe **article** não inicia uma nova página por omissão, enquanto **report** e **book** fazem.
twocolumn	Documento em duas colunas.
twoside, oneside	Especifica se a saída será de dupla face, ou simples. As classes **article** e **report** são de saída simples, e **book** de dupla face por padrão. Essa opção se aplica somente ao documento, e não faz nenhuma refência à impressora, de que é um documento dupla face, ou simples.
landscape	Altera o layout do documento para imprimir em modo paisagem.
openright, openany	Faz capítulos começarem apenas nas páginas do lado direito ou na próxima página disponível. Isso não funciona com a classe **article**, pois ela não prevê capítulos. A classe **report** por padrão começa capítulos na próxima página disponível e a classe **book** inicia nas páginas do lado direito.
draft	Sinaliza na margem direita do documento problemas com hifenização e justificação.

Exemplo de utilização das opções: `\documentclass[y1, y2, y3]{x}`, onde "x" é a classe, e o conjunto "y", são as opções. Apesar do exemplo ilustrar a utilização de mais de uma opção separada por vírgula, podemos usar somente uma, ou nenhuma (com a omissão dos colchetes).

4 Alguns pacotes

Para "habilitar funções" no LaTeX, utilizamos o que é denominado **pacote**. A inclusão de um pacote deve ser feita no preâmbulo, com o seguinte comando: `\usepackage[]{}`. Logo abaixo, vamos explorar algumas opções importantes que podem ser utilizadas nos exemplos desta apostila.

- **babel:** Usado para configurar a localidade, o que vai influenciar na hifenização. Exemplo de uso: (observe que colchetes servem para determinar a linguagem, ou dialeto)
  \usepackage[brazil]{babel}
  \usepackage[english]{babel}
- **color:** Usado basicamente para colorir texto e outros elementos (leia a subseção 5.3 para maiores informações)
- **graphicx:** Permite a inclusão de imagens no documento (leia a seção 6 para mais informações)
- **inputenc:** Configura a codificação do arquivo de entrada. Exemplo de uso: \usepackage[latin1]{inputenc} (permite o uso de acentuação, sem o uso de nenhum comando)
- **beamer:** Usado para criação de apresentações
- **hyperref:** Permite a criação de links para referência cruzada

Após escolher a "classe" do documentos e adicionar os pacotes que desejamos, podemos dar início ao documento por meio do comando \begin{document}. Feito isso, organizamos o documento por meio de seções, para isso, usamos \section{Nome da Seção}, essa seção será "válida" até o momento em que usarmos o comando \section{} novamente. Subseções são iniciadas com \subsection{} dentro de alguma seção existente.

5 Formatando o texto

Nesta seção iremos explorar alguns exemplos de comandos usados para editar áreas delimitadas no texto. Observe que o texto entre chaves, é o que será editado, e que as aspas não fazem parte do resultado nesses exemplos. Comandos relacionados ao tamanho do texto são diferenciados, sendo que o comando está dentro das chaves, juntamente com o texto a ser editado.

Os comandos podem ser usados em conjunto para que se tenha o resultado desejado, para isso, basta fazer como nesse exemplo: \comando1{\comando2{parâmetrocomando2}} (nesse caso, o parâmetro do comando 1 é a saída do comando 2).

5.1 Estilo

- \textit{itálico} resulta em "*itálico*";
- \textbf{negrito} resulta em "**negrito**";
- \textrm{romano} resulta em "romano";
- \textsf{sans serif} resulta em "sans serif";
- \textsc{caixa alta} resulta em "CAIXA ALTA";
- \texttt{máquina de escrever} resulta em "`máquina de escrever`".

5.2 Tamanho

- {\tiny o menor} resulta em "o menor";
- {\scriptsize muito pequeno} resulta em "muito pequeno";
- {\footnotesize quase pequeno} resulta em "quase pequeno";
- {\small pequeno} resulta em "pequeno";
- {\large grande} resulta em "grande";
- {\Large maior} resulta em "maior";
- {\LARGE maior ainda} resulta em "maior ainda".

5.3 Cores

Para usar essa funcionalidade é preciso adicionar os pacotes **graphicx** e **color** (\usepackage{graphicx, color}). Alguns exemplos de comandos que podem ser usados após adicionar os pacotes:

- \textcolor{blue}{texto em azul} resulta em "texto em azul";
- \textcolor{red}{texto em vermelho} resulta em "texto em vermelho";

- \textcolor{yellow}{texto em amarelo} resulta em "texto em amarelo";
- \textcolor{green}{texto em verde} resulta em "texto em verde".

Pode ser feito também da seguinte forma:

- {\color{red} texto em vermelho} resulta em "texto em vermelho".

5.4 Espaçamentos

- **Espaçamento Entre Linhas**

Para alterar o espaçamento entre linhas durante a criação de um documento em LaTeX, deve-se incluir o pacote `setspace` (\usepackage{setspace}). Com isso podemos usar espaçamento simples (singlespacing), espaçamento de 1.5 (onehalfspacing) e espaçamento duplo (doublespacing). Exemplos:

O código:

\onehalfspace

Tudo o que estiver escrito aqui, estará formatado com espaçamento 1.5 entre linhas.

\singlespace

Tudo o que estiver escrito aqui, estará formatado com espaçamento simples entre linhas.

\doublespace

Tudo o que estiver escrito aqui, estará formatado com espaçamento duplo entre linhas.

\singlespace

Resulta em:

Tudo o que estiver escrito aqui, estará formatado com espaçamento 1.5 entre linhas.

Tudo o que estiver escrito aqui, estará formatado com espaçamento simples entre linhas.

Tudo o que estiver escrito aqui, estará formatado com espaçamento duplo entre linhas.

O uso do comando \singlespace no final do exemplo serve para reconfigurar o tamanho normal de espaçamento para o restante do documento,

tendo em vista que esses comandos servem para todo o documento a partir do ponto em que são usados.

5.5 Caracteres especiais e acentuação

Em LaTeX, existem alguns caracteres especiais que são utilizados em comandos, por esse fato, esses caracteres não podem ser simplesmente digitados diretamente nos documentos, para resolver esse problema, basta usar "\" (barra invertida). Exemplos de caracteres especiais que devem ser digitados precedidos de \, para que sejam interpretados como texto e, consequentemente, aparecerem corretamente no documento: ˜, ˆ, #, $, %, &, {, }.

Para escrever barras invertidas de modo que apareçam no documento, devemos utilizar o comando: \textbackslash

Os comandos de acentuação devem ter {} no final, para especificar qual letra será acentuada (colocada entre as chaves), caso queira que apareçam sozinhos, deixe as chaves vazias. Exemplo: `\~{}`.

6 Inserindo figuras

Para inserir figuras é necessário a inclusão do pacote `graphicx`, da mesma forma como abordado em todo o restante deste manual (`\usepackage{graphicx}`). Os formatos de imagens suportados para essa operação são: PDF, JPG, e PNG.

Figura 1: Exemplo com Gaivota



A sintaxe básica de como inserir figuras está expressa logo abaixo:

1. \begin{figure}[!htb]

2. \centering

3. \includegraphics{nomedafigura}
4. \caption{Legenda}
5. \label{Rótulo}
6. \end{figure}

Na primeira linha dizemos que estamos iniciando um bloco para inserir a imagem (\begin{figure}), e passamos como opção (entre colchetes), !htb. Com isso, o próprio compilador irá escolher a melhor posição para a imagem. (h = here (aqui), t = top (topo), b = bottom (baixo)). Caso seja especificada somente uma dessas letras, (sempre com o "!" antes), a imagem ficará exatamente no lugar equivalente a essa letra.

Na segunda linha, especificamos que o alinhamento será centralizado (caso seja omitido, a figura fica alinhada à esquerda por padrão). Na terceira linha, definimos o nome da imagem a ser inserida (devemos passar o caminho completo da imagem, caso ela não esteja no diretório do projeto. Caso tenha uma pasta onde estão as imagens, e não queira digitar o caminho a todo momento, inclua no preâmbulo o comando: `\graphicspath{diretórioaqui}`). Para evitar problemas, não coloque a extensão do arquivo, pois essa pode estar com letras em caixa alta, ou baixa, e devem ser escritas exatamente como estão. Exemplo:
"imagem.png" é diferenciado de "imagem.PNG", use "imagem" para evitar problemas :)

Na quarta linha é especificada uma legenda para a imagem. Não é obrigatório colocar algo aqui, podemos simplesmente deixar como: `\caption{}`.

Na quinta linha, definimos uma identificação da imagem, usada, principalmente para referência cruzada, o que também é opcional. Na última linha, finalizamos o bloco de inserção de imagens.

***Observação:** A numeração usada antes de cada comando, é usada somente para facilitar a leitura, e não fazem parte dos comandos. Caso queira copiar um bloco inteiro desse manual, apague a numeração de linhas, caso essa tenha sido utilizada.

No momento de especificação da imagem que vamos incluir (\includegraphics{}), podemos passar algumas opções para customizar a imagem. A Tabela 3 apresenta essas opções e suas respectivas descrições:

Exemplo de uso: `\includegraphics[scale=0.5]{imagem}`. *Observação: Usar . (ponto) nos números reais e não inserir os valores entre aspas (valor e tamanho entre aspas somente para entendimento de que não são constantes).



Figura 2: GAIVOTA, nome vulgar extensivo a umas aves palmípedes da família dos Larídeos, algumas das quais comuns em Portugal, também conhecidas por alcatraz, bruto, galfoeira, falcoeira, etc.

Tabela 3: Opções do ambiente `figure` com suas respectivas descrições.

Opção	**Descrição**
width="tamanho"	Define a largura da imagem. Passar "tamanho" com a unidade. Ex: mm, cm.
height="tamanho"	Define a altura da imagem
keepaspectratio	Valor booleano (true/false), quando verdadeiro (true), mantém a proporção da figura, sem distorcê-la
scale="valor"	Redimensiona com base no "valor" passado. Ex: 0.5 = reduzir pela metade
angle="valor"	Rotaciona em graus a imagem (no sentido anti-horário)
page="valor"	Se a figura inserida é um PDF, "valor" define a página fonte

7 Criando tabelas

Para criar tabelas, o primeiro passo é delimitar a "área" na qual a tabela será criada, fazemos isso por meio de um ambiente do tipo `table`. Para iniciar (e terminar) um ambiente `table`, usamos os seguintes comandos:

1. \begin{table}[h]
2. \end{table}

No comando \begin{table}[h], estamos dizendo que estamos iniciando algo (begin), e logo após, o que estamos iniciando (table). O parâmetro h, passado entre colchetes, representa em que lugar colocaremos essa tabela, por hora, colocaremos "h" (here), indicando que ela ficará exatamente onde estamos criando.

Em \end{table}, estamos simplesmente falando que neste ponto, nossa tabela terminou, repare que `begin` e `end` serão usados para todas as estruturas que precisaremos.

Agora, iremos adicionar uma legenda a tabela, para isso, basta usar \caption{"um nome"}, dentro do ambiente tabela que criamos. Lembre-se que as aspas não fazem parte do comando.

Dentro do ambiente tabela, iremos criar outro ambiente, o `tabular`. Aqui especificamos o número de colunas, o alinhamento de cada uma, onde haverá linhas verticais separando colunas. Para criá-lo, basta usar \begin{tabular}{r|lr}. Nesse comando, observe que há parâmetros entre chaves depois do primeiro par de chaves.

O primeiro "r", nos diz que essa coluna será alinhada a direita, a barra (|) representa uma linha vertical, e "lr" duas colunas, uma alinhada a esquerda, outra a direita, sem linha vertical entre elas. O código a seguir mostra o que foi feito até agora:

```
\begin{table}[h]
\caption{Tabela Simples}
\begin{tabular}{r|lr}

\end{tabular}
\end{table}
```

Para colocar os dados, simplesmente escrevemos o campo, separamos da próxima coluna com o uso de um &, e quebramos a linha com \\. Pode parecer confuso somente com a explicação, observe o exemplo:

```
\begin{table}[h]
\caption{Tabela Simples}
\begin{tabular}{r|lr}
Número & Letra & Número \\
\hline % uma linha horizontal
1 & a & 1 \\
2 & b & 2 \\
\end{tabular}
\end{table}
```

Esse é o resultado do código:

Algumas vezes é preciso construir tabelas em que os campos devem

Tabela 4: Tabela Simples

Número	Letra	Número
1	a	1
2	b	2

ser mesclados. Nesses casos, podemos mesclá-las na horizontal e na vertical. Para mesclar colunas, usamos o comando: \multicolumn{"x"}{"y"}{"z"}, onde x é a quantidade de colunas que serão mescladas, y é o alinhamento do campo (c, l, r), e z o conteúdo do campo. Todos os parâmetros usados não devem ser informados com aspas.

Para mesclar linhas, usamos \multirow{"x"}{"y"}{"z"}, necessária a inclusão do pacote \usepackage{multirow} no preâmbulo, x representa a quantidade de linhas, y o tamanho do comprimento da coluna (cm, por exemplo), z o conteúdo do campo. Exemplo:

```
\begin{table}[h]
\center
\caption{Tabelas Com Campos Mesclados}
\vspace{0.5cm}
\begin{tabular}{| l | l | l |}
\hline
\multicolumn{3}{ |c| }{Colunas Mescladas} \\
\hline
Primeiro Campo & Segundo Campo & Terceiro Campo \\
\hline
\multirow{3}{*}{Linhas Mescladas}

& 10000 & 10000 \\
& 20000 & 20000 \\
& 30000 & 30000 \\
\hline
\end{tabular}
\end{table}
```

Resulta em:

Tabela 5: Tabelas Com Campos Mesclados

Colunas Mescladas		
Primeiro Campo	Segundo Campo	Terceiro Campo
Linhas Mescladas	10000	10000
	20000	20000
	30000	30000

8 Listas, índices, referências

- Listas

Os itens podem ser usados com marcações diferentes em listas. No caso da estrutura `itemize`, os itens são marcados com •.

Exemplo:
\begin{itemize}
\item Um item
\begin {itemize}
\item Um subitem
\end{itemize}
\end{itemize}

Resulta em:

- Um item
  - Um subitem
    - ...
      - ...

Outra forma, é utilizando a estrutura `enumerate`, que possui marcação com números, conforme o nível.

1. Os itens são representados com algarismos arábicos, no primeiro nível,
   (a) são representados com letras no segundo nível
      i. são representados com algarismos romanos no terceiro nível e
         A. são representados com letras maiúsculas no quarto nível.

- Índice - Conteúdo

Podemos criar vários tipos de índices automaticamente em LaTeX, iremos começar com o índice de conteúdo. Para criá-lo, basta usar esse comando: \tableofcontents. Observe que o índice será gerado no local que usamos o comando.

Geralmente usamos esse índice (Sumário) depois de criar o título, em um artigo (classe article) podemos criá-lo da seguinte forma:

\title{Título Aqui}
\author{Autor Aqui}
\maketitle

Podemos ainda quebrar a página após criar o sumário, usando:
\pagebreak.

- Índice - Figuras

Para gerá-lo, usamos o comando \listoffigures. Exemplo:

Lista de Figuras

- Índices - Tabelas

Para gerá-lo, usamos o comando \listoftables. Exemplo:

Lista de Tabelas

9 O ambiente matemático

Para incluirmos uma fórmula matemática em nosso documento, precisamos informá-la através do código fonte, da mesma forma como foi feito até então. Para que elas apareçam da forma correta, fazemos uso de alguns comandos dentro do "ambiente matemático" e devemos informar onde esse ambiente tem início e onde ele termina, por meio do caracter $.

Exemplo:

```
$ x=\frac{-b\pm\sqrt{b^2-4ac}}{2a} $
```

Resultado: $x = \frac{-b\pm\sqrt{b^2-4ac}}{2a}$

Nesse exemplo, a fórmula fica na linha, na posição em que foi digitada, no caso onde é necessário que essa fórmula esteja mais destacada, usamos dois $ para início, e dois para fim. Veja o resultado da mesma fórmula em destaque:

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

Os operadores que podem ser utilizados diretamente são:

```
+ - = < > / : ! ' | [ ] ( )
```

Apesar das chaves serem usadas para separar de forma lógica dentro do comando que usamos no ambiente matemático, elas não aparecem na fórmula do arquivo final, quando precisamos que elas apareçam, usamos o comando \{ ou \}.

- Expoentes

Para determinar que um valor será um expoente, fazemos os uso do caracter ˆ antes de informar o valor do expoente. Exemplo:

```
$x^2$
```

Resultado: x^2

- Índices

Para determinar que um valor será um índice, fazemos os uso do caracter _ antes de informar o valor do índice. Exemplo:

```
$x_i$
```

Resultado: x_i

Algumas observações: Valores de índices ou expoentes que possuem mais de um caracter, devem ser informados entre chaves. Quando um valor possui expoente e índice ao mesmo tempo, a ordem não afeta o resultado. Exemplo:

```
$x^{2n}_i$ equivale a $x_i^{2n}$
```

Resultado: x_i^{2n}

- Frações

Para frações simples, podemos simplesmente usar o caracter /, porém, numerador e denominador não ficarão um acima do outro. Exemplo:

```
$(a+b)/2$
```

Resultado: $(a+b)/2$

Para frações mais complexas, podemos usar o comando \frac. Exemplo usando a mesma fórmula anterior:

```
$\frac{a+b}{2}$
```

Reultado: $\frac{a+b}{2}$

- Raizes

Comando:

```
$\sqrt[3]{8}=2$
```

Resultado: $\sqrt[3]{8}=2$

Nesse exemplo, 3 representa o índice, 8 o radicando e 2 a raiz, caso o índice seja omitido, a raiz quadrada é gerada por padrão. Exemplo:

```
$\sqrt{4}=2$
```

Resultado: $\sqrt{4}=2$

- Somatório

Usamos o comando \sum. Exemplo:

```
$\sum_{i=1}^n a_i$
```

Resulta em: $\sum_{i=1}^{n} a_i$

- Integrais

    Usamos o comando \int. Exemplo:

```
$\int_{a}^b f(x)dx$
```

Resulta em: $\int_{a}^{b} f(x)dx$

- Coeficientes Binomiais

    Usamos através do comando \choose. Exemplo:

```
$$ {n+1\choose k}={n\choose k}+{n\choose k-1} $$
```

Resulta em:

$${n+1\choose k}={n\choose k}+{n\choose k-1}$$

Quando precisamos usar três pontos em nossas fórmulas, usamos o comando \ldots ou \cdots. Podemos usar também três pontos na vertical, com o uso do comando \vdots. Para que espaços sejam reconhecidos pelo compilador (que ignora espaços extras), podemos usar \quad, e caso seja necessário usar texto no meio equação, usamos \mbox{texto desejado}.

- Símbolos

Letras Gregas					
Símbolo	Comando	Símbolo	Comando	Símbolo	Comando
α	\alpha	β	\beta	γ	\gamma
δ	\delta	ϵ	\epsilon	ε	\varepsilon
ζ	\zeta	η	\eta	θ	\theta
ϑ	\vartheta	ι	\iota	κ	\kappa
λ	\lambda	μ	\mu	ν	\nu
ξ	\xi	o	o	π	\pi
ϖ	\varpi	ρ	\rho	ϱ	\varrho
σ	\sigma	ς	\varsigma	τ	\tau
υ	\upsilon	ϕ	\phi	φ	\varphi
χ	\chi	ψ	\psi	ω	\omega
Γ	\Gamma	Δ	\Delta	Θ	\Theta
Λ	\Lambda	Ξ	\Xi	Π	\Pi
Σ	\Sigma	Υ	\Upsilon	Φ	\Phi
Ψ	\Psi	Ω	\Omega		

Operadores Binários					
Símbolo	Comando	Símbolo	Comando	Símbolo	Comando
$\pm$	\pm	$\mp$	\mp	$\times$	\times
$\div$	\div	$\cdot$	\cdot	$\ast$	\ast
$\star$	\star	$\dagger$	\dagger	$\ddagger$	\ddagger
$\cap$	\cap	$\cup$	\cup	$\setminus$	\setminus
$\vee$	\vee	$\wedge$	\wedge	$\otimes$	\otimes
$\bigtriangleup$	\bigtriangleup	$\bigtriangledown$	\bigtriangledown	$\oplus$	\oplus
$\triangleleft$	\triangleleft	$\triangleright$	\triangleright	$\odot$	\odot
$\circ$	\circ	$\bigcirc$	\bigcirc	$\diamond$	\diamond

Relações					
Símbolo	Comando	Símbolo	Comando	Símbolo	Comando
$\le$	\le	$\ge$	\ge	$\sim$	\sim
$\not<$	\not<	$\not>$	\not>	$\neq$	\neq
$\subset$	\subset	$\supset$	\supset	$\approx$	\approx
$\subseteq$	\subseteq	$\supseteq$	\supseteq	$\simeq$	\simeq
$\in$	\in	$\notin$	\notin	$\equiv$	\equiv
$\perp$	\perp	$\propto$	\propto	$\cong$	\cong

Referências

http://www.dm.ufscar.br/profs/sadao/latex/tex-packages.php?lang=pt

http://www.mat.ufmg.br/~regi/topicos/intlat.pdf

https://aprendolatex.wordpress.com/2008/10/18/espacamento-entre-linhas-simples-duplo-e-de-15/

http://w3.ufsm.br/petmatematica/arquivos/minicurso_latex_2011.pdf

http://www.prograd.uff.br/estatistica/sites/default/files/Latex_net.pdf

http://www.icmc.usp.br/pessoas/francisco/SME0121/material/latex.pdf

http://latexbr.blogspot.com.br/2011/07/inserindo-figuras-no-latex.html

http://posgraduando.com/blog/como-fazer-tabelas-em-latex

http://www.vivaolinux.com.br/artigo/LaTeX-Figuras-e-tabelas?pagina=2

https://aprendolatex.wordpress.com/2008/10/22/fazer-uma-lista-de-numenclatura-notacao/